TECHNICAL REPORT RDMR-SS-11-35

VISUALIZATION AND ANALYSIS OF ARENA DATA, WOUND BALLISTICS DATA, AND VULNERABILITY/LETHALITY (V/L) DATA

Glenn E. Romanczuk, John M. Steen, David J. Bissell, and Joseph D. McSweeney
System Simulation and Development Directorate
Aviation and Missile Research, Development, and Engineering Center

February 2012

Distribution Code A: Approved for public release; distribution is unlimited.

DESTRUCTION NOTICE

DISCLAIMER

TRADE NAMES

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 074-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 2012	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Visualization and Analysis of Arena Data, Wound Ballistics Data, and Vulnerability/Lethality (V/L) Data	
6. AUTHOR(S) Glenn E. Romanczuk, John M. Steen, David J. Bissell, and Joseph D. McSweeney	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Commander, U.S. Army Research, Development, and Engineering Command ATTN: RDMR-SSM-G Redstone Arsenal, AL 35898-5000	8. PERFORMING ORGANIZATION REPORT NUMBER TR-RDMR-SS-11-35
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	A

13. ABSTRACT ***(Maximum 200 Words)***

This report discusses the development of tools to visualize arena data from tests that were set up to characterize munitions. The test data were collected in flash panels and bundles that only capture a small amount of data. The visualization tools allowed the analyst to recreate the test from the raw data and display key elements. Wound ballistics data were visualized and displayed in various Three-Dimensional (3-D) environments. Several innovative Python-based environments have been created to display Vulnerability/Lethality (V/L) in innovative ways. The environments utilized Python and open-source scenegraphs to assist in the understanding of the V/L data and analysis.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Visualization, Vulnerability/Lethality (V/L), Python, Arena Tests	21
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-**01**-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

There has been rapid growth in open-source toolkits and projects that center on immersive virtual environments and synthetic environments. This growth and availability has created an opportunity for the effectiveness, Vulnerability/Lethality (V/L), and the survivability of communities to leverage enormous capabilities which until now would have been major development effort. The growth in tools and software is similar to the growth in Constructive Solid Geometry (CSG) modeling and Computer-Aided Design (CAD) geometry modeling that happened over the preceding 20 years. When the Ballistic Research Laboratory (BRL)-CAD system was first introduced, it helped to simplify the creation of geometries of target vehicles for V/L studies. BRL-CAD is now an open-source tool and can be freely downloaded from the web. The ability to display, edit, and render complex CSG targets and prepare them for detailed analysis in many fields was an enabling technology that set a high standard and helped to create the standards that Department of Defense (DoD) professionals in effectiveness, V/L, and survivability continue to depend upon today. [1]

This report covers the emergence of a set of tools and toolkits that share many key features. The primary features allowed many detailed geometries to be placed in an environment that can be as detailed as the modelers, artists, and engineers can envisage. Typically, the scene has a detailed scenegraph which manages the relationships between geometric objects and elements. They also usually contain an immersive characteristic that allows multiple "windows," or views, into the environment from multiple perspectives. Additionally, they contain the ability to visualize and animate the elements in the scene based on a frame rate or a time-based manner. Most importantly, these are not just animations for artistic purposes but rather highly detailed engineering simulations that incorporate detailed physics models. The detailed toolkit of physics-based V/L routines has been developed and configured to be called utilizing the Python scripting language. A number of these tools can input simulation data output by other simulations of Army missile flyout trajectories which are generated at Aviation and Missile Research, Development, and Engineering Center (AMRDEC) using sophisticated methodologies. The toolkit consists of modules including raytracing, wound ballistics, penetration, trajectory routines, and other special routines for V/L. The majority of these will not be covered in this report. This report builds upon the previously reported efforts in Reference 2.

II. SCENEGRAPH-BASED TOOLS

A. ProspectV2, V2.4, and ProspectV3

The ProspectV2 synthetic and Virtual Environment system and architecture was developed at AMRDEC by Dr. Doug Meyer. A number of similar tools focusing on Virtual Reality (VR) concepts were developed during Dr. Meyer's tenure in the Virtual Reality Laboratory (VRL) at the University of Alabama in Huntsville (UAH). The VRL was funded by AMRDEC. Dr. Meyer greatly enhanced the Prospect system by utilization of the Python interpreter and the transition of the tools and concepts to the company he founded, Envisage, Inc.

Figure 1 shows the architecture of the ProspectV2 system with the integrated Python interpreter. This baseline architecture has Two-Dimensional (2-D) and Three-Dimensional (3-D) Graphical User Interface (GUI) routines, Synthetic Environment routines that include faceted raytrace, communications support to allow immersion on multiple machines, or data input from other simulations.

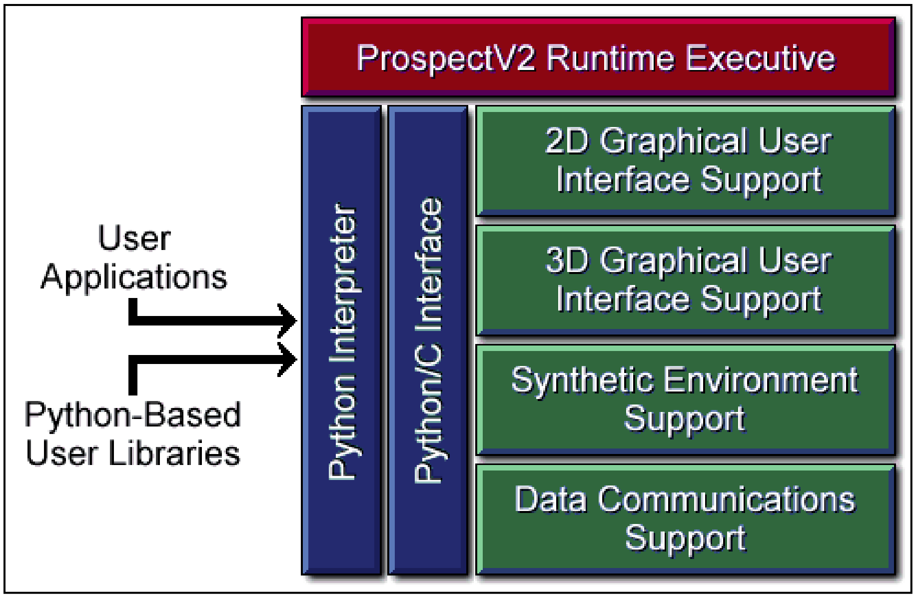

Figure 1. Overview of ProspectV2 [2]

ProspectV2.4 was developed with the help of Simulation Technologies (SimTech) Inc. SimTech updated the version of Python to both Python 2.1 and 2.4. This update allowed the package to grow and allowed new detailed physics-based packages to use emerging Python techniques in the ProspectV2 system. The system had been constrained to utilize the Python 1.5 interpreter and needed to be upgraded to utilize the expanding power of Python.

ProspectV3 was created by Gleason Research Associates by changing the architecture to allow the Prospect-based scenegraph and methods to be called, as the now more traditional site-packages. GUI development has also been improved by allowing backward compatibility to the early Prospect calls, while adding support for both Tkinter (Tk) and WxPython. The limitation of total facet count was removed, allowing much larger models to be utilized up to approximately 4 billion elements. New features include the ability to load native wafefront geometry file formats (OBJ files), in addition to Prospect's geometry (ODL data) file format. This version of Prospect retains the capability for cross-platform compatibility and should run on various systems that have Python without modifying the script [3].

1. Fragment-Pattern Visualization

There are many times when the only information that is needed is a correct representation of where fragmentation patterns will hit targets or elements that would filter out the fragmentation effects. The tools that have been developed use both scenegraph or game mode trajectory equations that can move nodes based on velocity and acceleration. This is utilized typically for slow moving objects. However, high-speed objects need trajectory routines or augmented scenegraph methods to make sure that the accuracy needed is obtained. Specialty trajectory modules have been written and integrated into the Military Operations in Urban Terrain (MOUT) lethality toolkit that can be called from Python and the majority of tools discussed in this report. Figure 2 shows a typical tool that has been created. Geometry methods also allow the call to high-detail wound ballistics modules after the location of impact and path of the fragment has been determined.

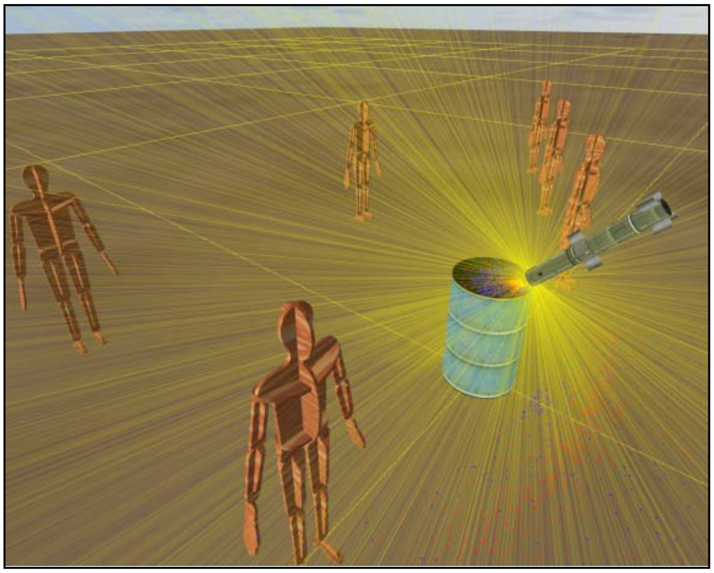

Figure 2. Fragment-Shotline Tools

2. Arena Tools

Arena testing is an important part of Weapon System Development (WSD). This testing helps to determine which warhead has the ability to meet the designer's requirements for lethality and effectiveness. There are many types of arena tests that are conducted by munitions designers and warhead engineers. Some of these tests are geared toward safety if a missile detonates early as it travels toward a target. Figure 3 shows a traditional arena with collection media and flash panels.

Figure 3. Typical Arena Setup

Detailed simulations of arena tests have been created. There are many types of analysis done with the virtual arenas, and the tools make use of the immersive quality of the environments. One use of this type of simulation is to compare and convert raw arena test data into the final arena test data file. Once a file is created, it can be compared visually to other data files or to other available data. Figure 4 shows a view of a standard arena with sample data indicated on transparent collection bundles.

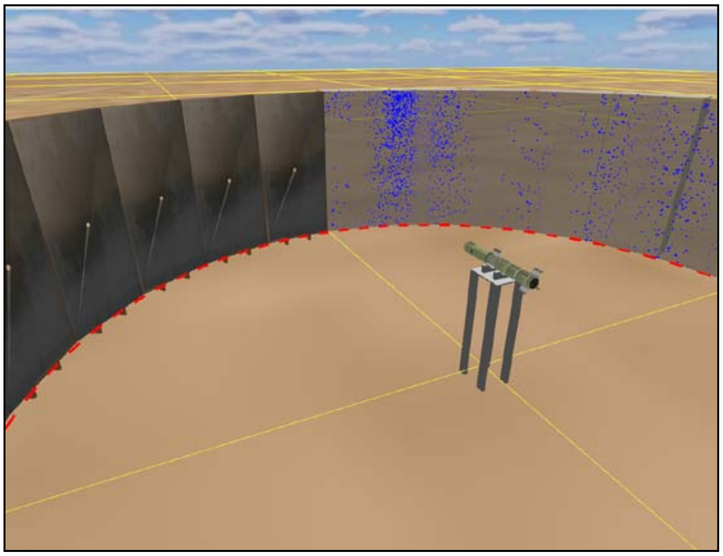

Figure 4. Virtual Arena Visualization

Live-fire arenas are also a type of arena test that can use the abilities of the software tools that have been developed. The arenas in this type of testing can be either circular metal arenas fully enclosed with floors and all panels labeled or larger arrays of metal witness panels on arc at several radii. The Institute for Defense Analysis (IDA) has utilized this approach on the Guided Multiple Launch Rocket System (GMLRS) Unitary warhead program (Figure 5). According to IDA, this approach allows researchers to "unobtrusively gather realistic fragment data" [4]. The live-fire test data is then compared to detailed computer simulations that simulate the endgame encounter and determine if the modeling for lethality can be verified.

Figure 5. GMLRS Unitary Live-Fire Arena [4]

The AMRDEC MOUT team has created various test setups from dynamic arena events and has used the MOUT lethality toolkit and Prospect-based environments to evaluate fragmentation patterns collected from live test events with the simulations of those same events. Work is ongoing to enhance the underlying fragment-generation schemes to more closely match weapons when they differ from more conventional weapons. Figure 6 shows a virtual arena test setup that is very similar to the one utilized and advocated by IDA.

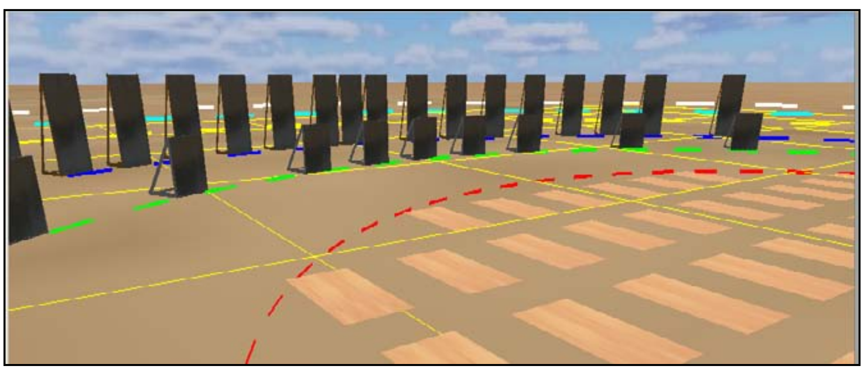

Figure 6. Virtual Live-Fire Arena Panel Layout

3. VSUV and HVTKill

View Sport Utility Vehicle (VSUV) and High-Value Target Kill (HVTKill) are tools that add in several penetration methodologies to be able to filter fragments as they pass through barrier materials. These fragment-penetration routines draw upon DoD standard penetration algorithms that have been modified to allow them to be called from Python programs.

This type of tool builds upon the other capabilities that have been illustrated in Prospect. Fragment generation occurs at the point of detonation; raytrace and trajectory methods fly out the fragments and determine intersections with the nodes in the environment. These hitpoints are then filtered to see which fragments penetrate the barriers along the shotline. Finally, if critical components are impacted, the appropriate module is called to determine the level of damage of that component. The tools have the ability to turn on key visualization aspects, such as the two fragment materials color coded in Figure 7. Additionally, the user can immerse in the scene to review and understand the complex interactions that are being simulated.

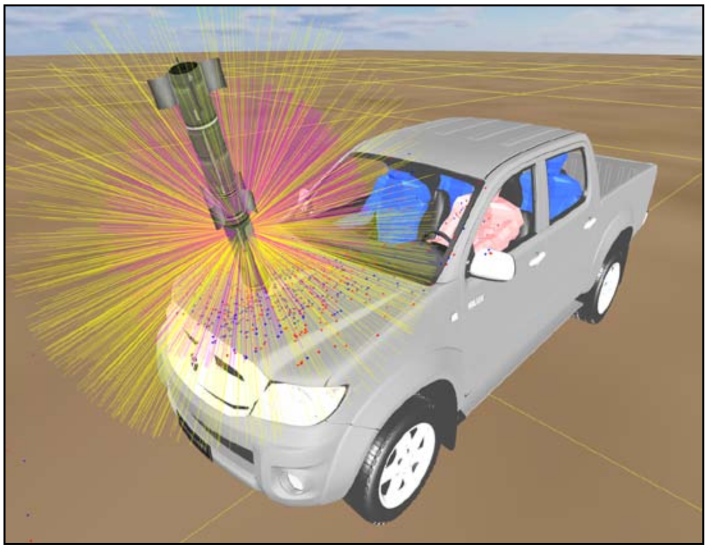

Figure 7. HVTKILL with Standard Truck

B. PANDA

The Platform Agnostic Networked Display Architecture (PANDA) 3-D simulation engine is a feature-rich toolkit that provides rendering support based on OpenGL or DirectX on many different types of platforms. These include Linux, Windows, Silicon Graphics, Inc. (SGI), and Sun. These are the same platforms that ProspectV2 Python scripts can run on without change to the script. PANDA3-D was not created with scientific or engineering simulation in mind. In fact, it was originally developed for Walt Disney. This engine was made open source in 2002 in order to collaborate more effectively with universities and government laboratories. PANDA's lineage can be traced back to several innovative VR systems developed at the Disney VR studios and is still used in Disney Imagineering [5].

There are many similarities in overall design between PANDA and Prospect; however, the implementation of many of the features is quite different. PANDA contains a number of scenegraph and node manipulation resources that are similar in type to the Synthetic Environment (SE) Tree functions in ProspectV2 and ProspectV3. PANDA3-D also has a 2-D GUI capability by using the standard Python interface to the Tk GUI toolkit and Python Mega Widgets. This makes it similar to ProspectV3. PANDA3-D also contains an abundant collision detection capability. This capability is geared toward gaming, but highly-detailed engineering level data can be pulled from the collision calls. This can also be augmented by BRL-CAD raytrace with a module from the MOUT lethality toolkit. In the following section, a detailed example of software based on PANDA3-D concepts is examined [5].

1. Recoil

Recoil is the name of a software tool created by Gleason Research Associates under a contract from the U.S. Army Research, Development, and Engineering Command (RDECOM). AMRDEC created the general outline and requirements for this tool. Several critical methodologies were provided to support this effort. The selection of PANDA3-D allows for the majority of features that are needed for this overall effort in scenegraph and visualization support utilizing Python. Figure 8 shows the system overview.

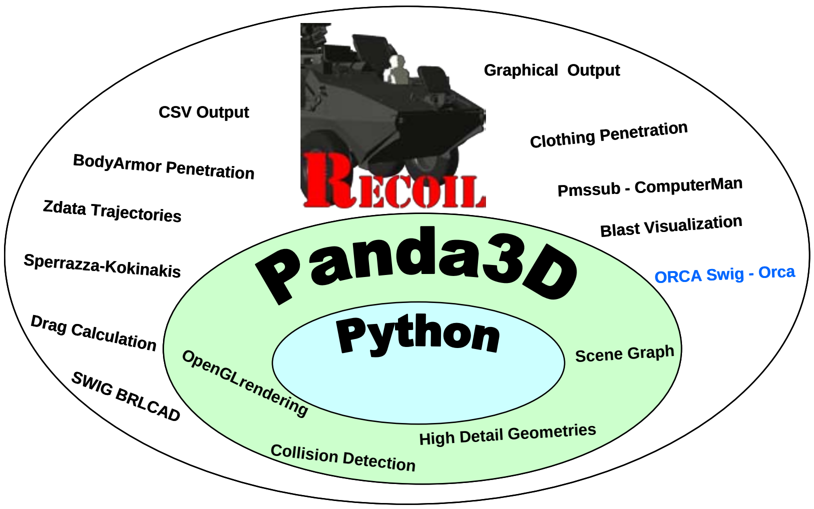

Figure 8. The PANDA-Based Recoil Collateral Hazard Tool

The Recoil tool was specifically created to determine collateral hazard calculations using high-detail wound ballistics models and proper trajectories for debris and fragments. The image in Figure 9 shows weapon-debris data, fragment data, and blast incapacitation estimates, all surrounding a protected vehicle. The user can immerse in the data that has been run and explore the incapacitation level of each nearby combatant or non-combatant. The tool also utilizes standard Active Protective System (APS) simulation output to place the interceptor and threat. A manual method is also provided.

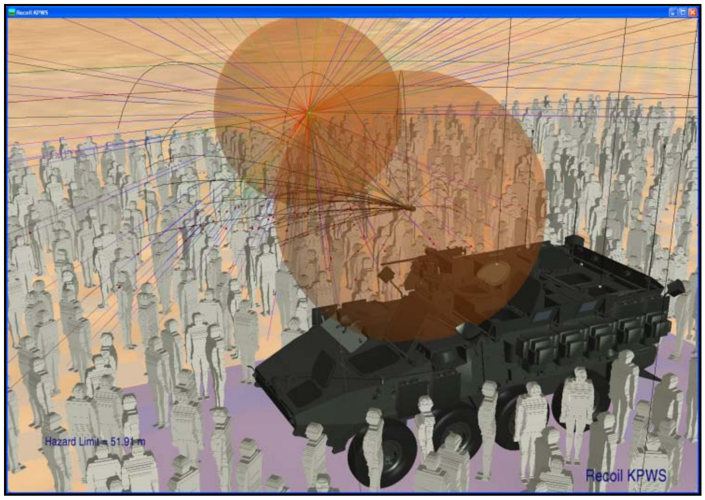

Figure 9. The Recoil Collateral Hazard Model

2. A2Z

The A2Z project is both a tool to create Zdata files from standard arena data collected from tests and a method to compare data processed with current Zdata files. The immersive tool uses the PANDA environment and allows checking of the data input and the final data. The tool is similar to the Arena Data visualization in the Prospect section but adds the ability to process high-speed video data that helps to calculate the initial velocity of the fragments for each zone. The zones are clearly visible in Figure 10. This tool is being developed by Gleason Research Associates and integrates several modules developed by the MOUT team into the final tool. The ideas were then implemented into a PANDA-based tool.

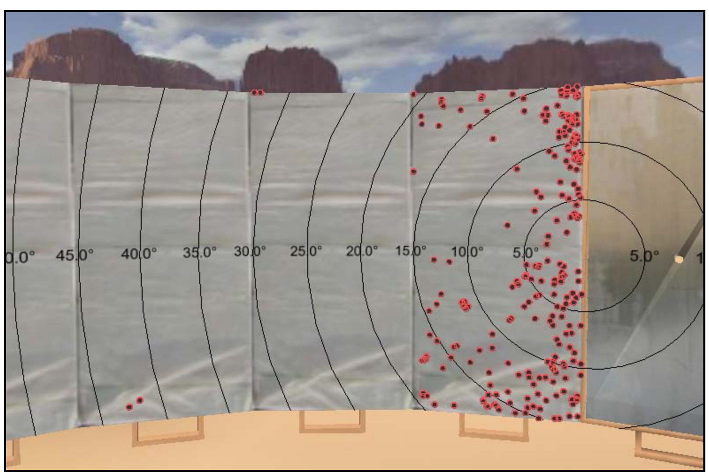

Figure 10. The A2Z Zdata Creation and Comparison Tool

C. Delta3-D

Delta3-D is an OpenSceneGraph (OSG)-based gaming, simulation, and virtual environment. The developers of Delta3-D have determined that many DoD simulation projects end up being very constrained by the choice of system architecture and of contractor that is selected to develop simulations and tools. Delta3-D addresses these concerns by adopting an open-source focus with a top-level support Application Programming Interface (API) that allows the integration of very powerful open-source tools. Delta3-D provides for Python as the scripting interface and provides full Python binding to the entire package. This is a key element in Delta3-D's utility.

Figure 11 shows the layout of the tool and several key elements [6].

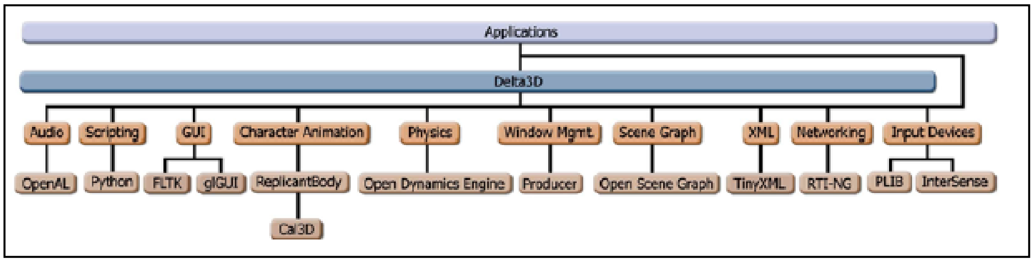

Figure 11. Delta3-D Overview [6]

Delta3-D uses OSG for rendering. As can be seen in Figure 11, several useful tools for military game and simulations are integrated, including Open Dynamics Engine (ODE) and Cal3-D for physics and character animation. Several of these tools are primarily geared at training and gaming type application but can be harnessed for our needs. However, the important thing to understand is that the motivations that drove the creation of this framework and overall architecture are similar in nature to the forces that have shaped all of the tools that have been discussed. The Personnel V/L tools and MOUT tools that are being developed at AMRDEC share many key features with efforts like Delta3-D but on a much smaller scale [6].

D. MOUTEndgame

MOUTEndgame has also been though several versions. The first tool was written in C with Microsoft Foundation Class (MFC) as the primary library to utilize for the GUI. However, that effort had serious limitations. A Prospect-based tool was then created and served as a platform to prototype the ideas needed in linking simulation data with MOUT lethality data.

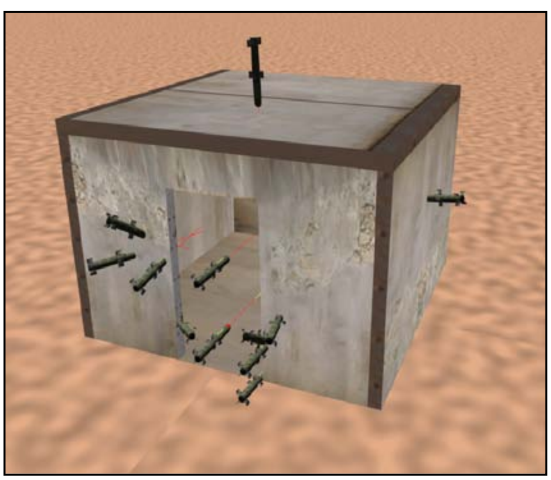

Figure 12. A MOUT Target in MOUTEndgame

The current version of MOUTEndgame utilizes the Delta3-D Virtual environment but shares many features with the previous Prospect-based tool. The user can read and visualize the impact points on the selected target. The detonation point is then determined from several input curves and values. This detonation point is then tied in with pre-computed 3-Dpimms lethality (Figure 13) data, computed, and stored in a 3-DPI file. This file format is tied to the detonation point.

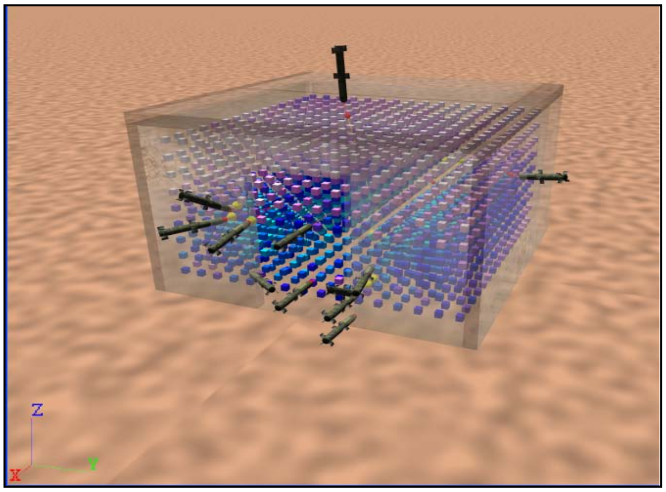

Figure 13. 3-DPimms Data Visualized

D. Navy ETB

The Navy Effectiveness Toolbox (ETB) effort has created a remarkable set of tools with most of the features of standard V/L assessment methodologies while incorporating many of the graphical and visualization features that are found in Prospect, PANDA3-D, and Delta3-D. Naval Surface Warfare Division Dahlgren integrated a wide variety of features into an OSG-based C++ approach. Figure 14 shows some of the elements that have been brought into the ETB system. An interesting point of departure from Prospect, PANDA3-D, and Delta3-D is the use of a Windows-based GUI interface and the use of standard object oriented C++.

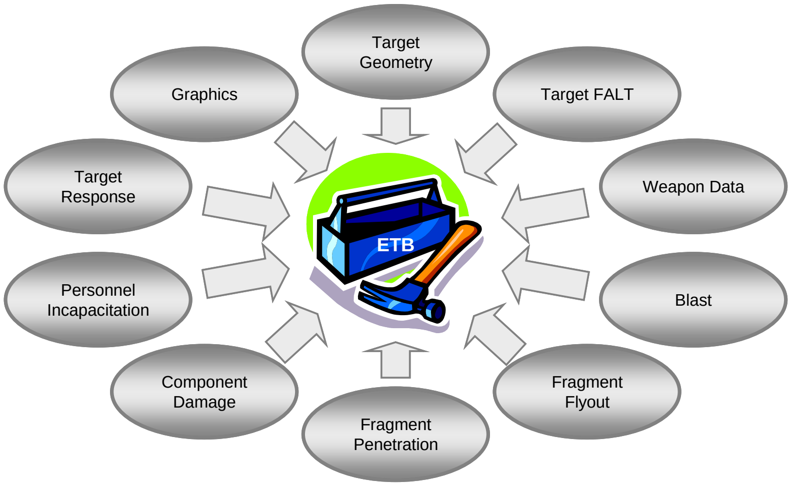

Figure 14. ETB [7]

Some of the basic visualization functionality of ETB was used in a Safe Separation analysis. Key elements were discussed and a movie clip of warhead view data at the 50th National Defense Industrial Association (NDIA) Fuze Conference was displayed. This is extraordinary work for a small number of dedicated programmers who have integrated key software modules into a useful framework. This work, and several of the key elements that have been presented in this report, would serve the Army Research Laboratory (ARL) well as it embarks on modification of its Modular Unix-based Vulnerability Estimation Suite (MUVES) lethality code [7].

III. VISUALIZATION

This tool uses Python and raytracing but does not need the overhead of a full scenegraph. In fact, even though rendering the scene in 3-D and then developing 2-D images would be easy to do with the tools described in this report, the decision was made to render images to 2-D and then present those images, allowing simulation data to be merged into the tool without the need for projections that could distort the underlying data.

WxSimVu is a new version of tools previously developed at the UAH Visualization and Simulation Laboratory. The new WxSimVu GUI can be seen in Figure 15. It is a primary tool for small weapons that approach the target from viewpoints around the target and not the typical encounters found in air-to-air engagements or those of highly-maneuverable missiles and air-to-ground missiles. New techniques have been coded that allow for autocreation of both grids that represent the target and line drawings based on BRL-CAD raytrace of the selected targets.

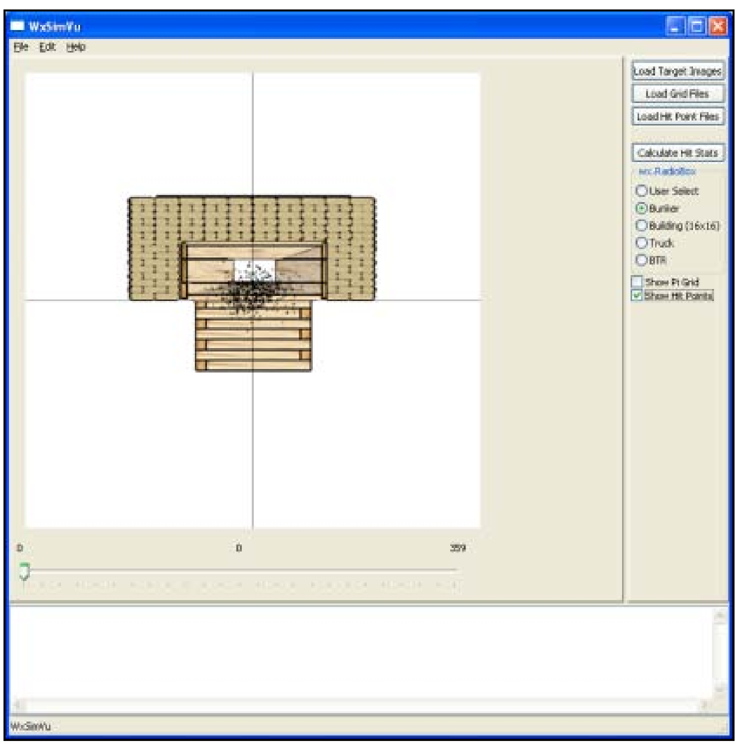

Figure 15. WxSimVu Displays a Standard Bunker

REFERENCES

1. http://brlcad.org/

2. Romanczuk, G., Pike, A., and Crow, E., "Integration of Detailed Personnel Injury Methodologies into Immersive Virtual Environments," 5th AIAA Biennial National Forum on Weapon System Effectiveness, AIAA, October 2007.

3. Sloan, S. and Randall, R., "MOUT Tool and WxPython Implementation Update Final Report," Gleason Research Associates, September 2009.

4. https://www.ida.org/researchareas/systemevaluations/testandevaluation.php

5. Goslin, M., and Rose, D., et. al., "The Panda3D Engine - Python Scripting for Game and Simulation Development," http://www.python.org/pycon/dc2004/papers/29

6. McDowell, P., Darken, R., Sullivan, J., and Johnson, E., "Applications, Methodology, Technology," The Journal of Defense Modeling and Simulation, The Society for Modeling and Simulation International, Vol. 3, No. 3, pp. 143–154, July 2006.

7. Knott, B., "Computer Software Description." ETB Overview, November 2006.

LIST OF ACRONYMS AND ABBREVIATIONS

2-D	Two-Dimensional
3-D	Three-Dimensional
AMRDEC	Aviation and Missile Research, Development, and Engineering Center
API	Application Programming Interface
APS	Active Protective System
ARL	Army Research Laboratory
BRL	Ballistic Research Laboratory
CAD	Computer-Aided Design
CSG	Constructive Solid Geometry
DoD	Department of Defense
ETB	Effectiveness Toolbox
GMLRS	Guided Multiple Launch Rocket System
GUI	Graphical User Interface
HVTKill	High-Value Target Kill
IDA	Institute for Defense Analysis
MFC	Microsoft Foundation Class
MOUT	Military Operations in Urban Terrain
MUVES	Modular Unix-based Vulnerability Estimation Suite
NDIA	National Defense Industrial Association
OBJ	A Wavefront geometry file
ODE	Open Dynamics Engine
ODL	A Prospect geometry file
OSG	Open Scene Graph
PANDA	Platform Agnostic Networked Display Architecture
RDECOM	Research, Development, and Engineering Command
SE	Synthetic Environment
SGI	Silicon Graphics, Inc.
SimTech	Simulation Technology, Inc.
Tk	Tkinter
UAH	University of Alabama in Huntsville
V/L	Vulnerability/Lethality

LIST OF ACRONYMS AND ABBREVIATIONS (CONCLUDED)

VR	Virtual Reality
VRL	Virtual Reality Laboratory
VSUV	View Sport Utility Vehicle
WSD	Weapons System Development

INITIAL DISTRIBUTION LIST

		Copies
Weapon Systems Technology Information Analysis Center Alion Science and Technology 201 Mill Street Rome, NY 13440	Ms. Gina Nash gnash@alionscience.com	Electronic
Defense Technical Information Center 8725 John J. Kingman Rd., Suite 0944 Fort Belvoir, VA 22060-6218	Mr. Jack L. Rike jrike@dtic.mil	Electronic
AMSAM-LI	Ms. Anne C. Lanteigne anne.lanteigne@us.army.mil	Electronic
	Mr. Michael K. Gray michael.k.gray@us.army.mil	Electronic
RDMR		Electronic
RDMR-CSI		Electronic
RDMR-SSM-G	Mr. David Bissell david.j.bissell@us.army.mil	Electronic
	Mr. Glenn Romanczuk glenn.romanczuk@us.army.mil	Electronic/Hardcopy
	Mr. John Steen john.m.steen@us.army.mil	Electronic
RDMR-WDG-R	Dr. Brian Smith brian.jennings.smith@us.army.mil	Electronic
U.S. Army Test and Evaluation Command 4120 Susquehanna Ave. APG, MD 21005-3013	Ms. Kathy Fontaine kathy.fontaine1@us.army.mil	Electronic
	Mr. Matthew Pandullo matthew.pandullo@us.army.mil	Electronic
U.S. Army Research Laboratory 328 Hopkins Road APG, MD 21005-5068	Mr. Gregory Dietrich gregory.k.dietrich@us.army.mil	Electronic
	Mr. David Fordyce david.fordyce1@us.army.mil	Electronic
	Mr. William Mermagen bill.mermagen@us.army.mil	Electronic
	Mr. Richard Pearson richard.j.pearson@us.army.mil	Electronic
U.S. Army Materiel Systems Analysis Activity	Mr. Michael Booth-Neades michael.boothneades@us.army.mil	Electronic

INITIAL DISTRIBUTION LIST (CONCLUDED)

		Copies
	Mr. Matthew Rosenblatt matt.rosenblatt@us.army.mil	Electronic
OSD, DOT&E Live Fire 1700 Defense Pentagon Washington, D.C. 20301-1700	Ms. Nicole Brockhoff nikki.brockhoff@us.army.mil Ms. Stephanie Koch stephanie.koch@osd.mil	Electronic Electronic
Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311	Dr. Joel Williamsen joel.williamsen@us.army.mil	Electronic